Implementation of Adaptive Digital Controllers on Programmable Logic Devices

David A. Gwaltney, Kenneth D. King and Keary J. Smith

NASA Marshall Space Flight Center, Huntsville, AL

Abstract

Much has been made of the capabilities of Field Programmable Gate Arrays (FPGA's) in the hardware implementation of fast digital signal processing functions. Such capability also makes an FPGA a suitable platform for the digital implementation of closed loop controllers. Other researchers have implemented a variety of closed-loop digital controllers on FPGA's. Some of these controllers include the widely used Proportional-Integral-Derivative (PID) controller, state space controllers, neural network and fuzzy logic based controllers. There are myriad advantages to utilizing an FPGA for discrete-time control functions which include the capability for reconfiguration when SRAM-based FPGA's are employed, fast parallel implementation of multiple control loops and implementations that can meet space level radiation tolerance requirements in a compact form-factor. Generally, a software implementation on a Digital Signal Processor (DSP) device or microcontroller is used to implement digital controllers. At Marshall Space Flight Center, the Control Electronics Group has been studying adaptive discrete-time control of motor driven actuator systems using DSP devices. While small form factor, commercial DSP devices are now available with event capture, data conversion, Pulse Width Modulated (PWM) outputs and communication peripherals, these devices are not currently available in designs and packages which meet space level radiation requirements. In general, very few DSP devices are produced that are designed to meet any level of radiation tolerance or hardness. An alternative is required for compact implementation of such functionality to withstand the harsh environment encountered on spacecraft. The goal of this effort is to create a fully digital, flight ready controller design that utilizes an FPGA for implementation of signal conditioning for control feedback signals, generation of commands to the controlled system, and hardware insertion of adaptive control algorithm approaches. Radiation tolerant FPGA's are a feasible option for reaching this goal.

I. INTRODUCTION

The use of programmable logic devices, specifically Field Programmable Gate Arrays (FPGA's) for the implementation of digital signal processing functions has been heavily promoted and extensively studied[1,2]. There are many examples of the implementation of high order filters for radar, sound processing, and general Finite Impulse Response (FIR) filter implementation[3,4,5,6]. The capabilities of FPGA's that make them attractive for general digital signal processing applications also make them attractive for the implementation of digital controllers. A digital filter very close to, if not exactly, the form of an Infinite Impulse Response (IIR) filter can represent most digital controllers.

Adaptive control is a technology that is used for control of systems having dynamics, which vary over time or with operating conditions. In it's broadest definition, adaptive control includes controllers with gain scheduling, self-tuning controllers and model reference adaptive control. This work is primarily concerned with the self-tuning regulator due to Astrom and Wittenmark[7]. The structure of this adaptive controller is illustrated in Figure 1. Self-tuning regulators combine a linear controller with a parameter identification approach to provide a structure in which the gains of the controller are calculated on-line. Applications of interest are electromechanical actuators (EMA's) currently in use on spacecraft, and those proposed for a power-by-wire system. Some examples are aero surface positioning control, thrust vector control, valve positioning, motor-driven pumps and translation systems for space-borne experiments.
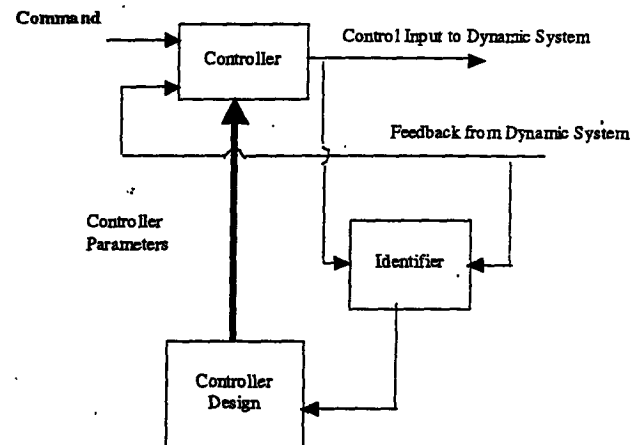

Figure 1 Diagram of the self-tuning regulator

Open literature and current observation indicate the vast majority of EMA controllers in industrial and aerospace applications employ a linear control approach with fixed gains[8,9,10,11,12]. It's generally recognized that the majority of deployed controllers in industrial settings are poorly tuned due to lack of skilled personnel or lack of application of analytical approaches[7]. In controllers with fixed gains, even analytically derived tunings may not be optimal due to uncertainty in system parameters, i.e. inertia, damping or load. These controllers cannot accommodate changes in actuator dynamics due to wear and tear and operation at loads outside the expected range. On-line

adaptive control addresses these shortcomings and increases the "intelligence" of a closed loop control system. Adaptive controllers can provide improved control system performance in the face of unanticipated changes in actuator/mechanical system dynamics and allows self-tuning of actuator control loops. System parameter identification employed by the adaptive controller can be used in a fault-detection and isolation scheme.

Actuator or subsystem-level digital controllers are frequently implemented using digital signal processors (DSP's). DSP's are a good platform for digital control because they are designed to perform repetitive, math intensive operations. Manufacturers such as Texas Instruments [13], Analog Devices [14] and Motorola [15] are producing mixed-signal DSP devices for instrumentation and control applications that include peripherals for analog-to-digital conversion, event-capture, quadrature signal decoding, PWM outputs and serial communications. Implementation of adaptive control on DSP's has been performed in a laboratory environment to study adaptive control approaches for the control of motor driven actuators[16]. The drawback to this approach is the very limited availability of DSP's produced to be radiation hard or tolerant. None of the new mixed signal devices are available in such packages, and the availability of these devices in military packages is extremely limited. This limits the type of space missions on which such devices can be used.

An alternative platform for implementation is required. The implementation of digital controllers on FPGA's has been reported extensively. Some examples include, a PID controller for wheel speed control as part of a digital controller for a wheelchair [17], implementation of controllers for robotic applications[18], direct torque control of an induction motor[19] and Implementation of a Kalman Filter and Linear Quadratic Gaussian controller applied to control of an inverted pendulum[20][21]. There are also examples of the implementation of controllers utilizing soft computing techniques, such as neural network implementation for control of an induction motor[22], and a Fuzzy logic controller for a variable speed generator[23]. In the referenced papers, the FPGA approach for implementation of digital controllers is selected because FPGA's can provide reconfigurable hardware designs, can process information faster than a general purpose DSP, can allow the controller architecture to be optimized for space or speed and bit widths for data registers can be selected based on application needs. Additionally, implementation in VHDL or Verilog allows the targeting of a variety of commercially available FPGA's. Implementation of digital controllers in FPGA's for space applications is attractive because FPGA's are available in radiation tolerant packages, and they allow complex, digital control operations and controller interface peripherals to be contained in a compact form factor. Further, multiple digital control loops in one FPGA can replace analog control loops implemented in many space consuming and power hungry radiation tolerant analog integrated circuits.

This paper will outline the implementation of digital controllers, both linear and adaptive to provide a basis for understanding the issues with implementation in an FPGA. The experimental system to be controlled will be presented, along with the proposed structure for implementation of an adaptive controller in an FPGA.

II. CONTROLLER IMPLEMENTATION

A. Digital Controllers

In general, digital controllers can be implemented as digital filters in the following form[24], where k is the current sample in time, for a given sample period T;

$$y(k) = \sum_{i=0}^{n} a_i x(k-i) - \sum_{i=1}^{n} b_i y(k-i)$$

In this form y(k) is the output, x(k) is the input, a_i and b_i are coefficients, or gains, of the controller. These gains must be selected to produce the desired controller response for a given dynamic system to be controlled. The process of determining these gains is called "tuning." The structure of this digital controller is illustrated in Figure 2. In the figure, the Z^{-1} blocks represent delays of one sample period.
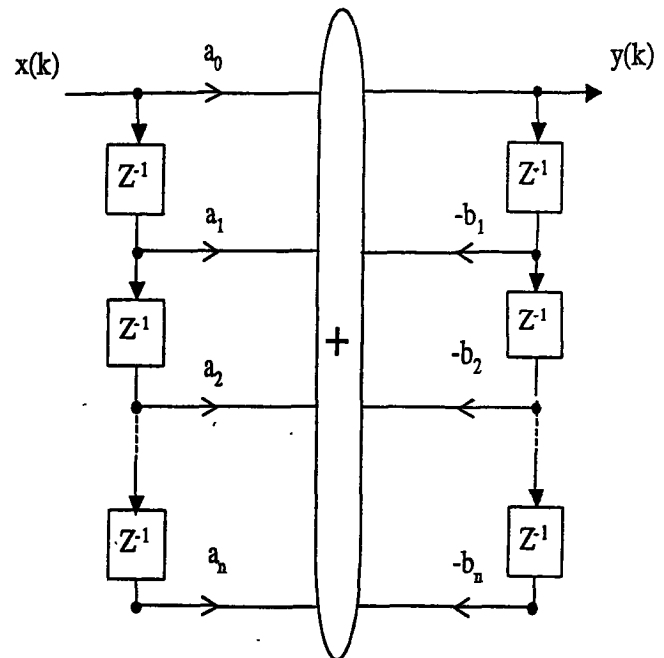

Figure 2 Diagram of the digital controller[24]

When n = 2, a second order filter is obtained which can be used to implement second order controllers or cascaded to create higher order controllers. This representation is shown in the sampled time domain below;

$$y(k) = a_0 x(k) + a_1 x(k-1) + a_2 x(k-2) - b_1 y(k-1) - b_2 x(k-2)$$

The z-transform of this gives the following transfer function;

$$D(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

B. PID controller

The well-known PID controller can be implemented using a second order digital filter. The continuous time representation of a PID controller is as follows;

$$u(t) = K_P e(t) + \frac{1}{(K_I)} \int e(t) dt + K_D \frac{de(t)}{dt}$$

In this controller, K_P is the proportional gain, K_I is the integral gain, K_D is the derivative gain, and e(t) is the error between the desired response and the actual system response. In the sampled time domain, with sample period T, the PID controller is represented as[24]

$$u(k) = K_P e(k) + \frac{T}{K_I} S(k) + K_D \frac{e(k) - e(k-1)}{T}$$

$$S(k) = S(k-1) + \frac{T}{2}[e(k) + e(k-1)]$$

The z-transform of this controller gives the following transfer function;

$$D(z) = K_P + \frac{K_I T}{2}\left(\frac{z+1}{z-1}\right) + \frac{K_D}{T}\left(\frac{z-1}{z}\right)$$

With some manipulation this transfer function can be represented as a second order filter;

$$D(z) = \frac{(K_P + K_I T/2 + K_D/T) + (-K_P + K_I T/2 - 2K_D/T)z^{-1} + K_D/Tz^{-2}}{1 - z^{-1}}$$

$$a_0 = K_P + \frac{K_I T}{2} + \frac{K_D}{T} \qquad a_2 = \frac{K_D}{T} \qquad b_1 = -1$$

$$a_1 = -K_P + \frac{K_I T}{2} - \frac{2K_D}{T} \qquad\qquad\qquad b_2 = 0$$

C. Self-Tuning Controller

For this work, the Self-Tuning Regulator, due to Astrom and Wittenmark [7], will be referred to as a Self-Tuning Controller. "Regulation" implies the rejection of disturbances to maintain a controlled process at a constant setpoint, while "control" implies the broader action of following a desired trajectory and rejecting disturbances. This controller makes use of a general linear controller, shown as a z-transform below, where u_c is the desired trajectory, y is the controlled process output and u is the control input to the process.

$$R(z)U(z) = T(z)U_c(z) - S(z)Y(z)$$

The coefficients of R, T and S are calculated using pole placement design to produce the desired closed-loop response. A diagram of a closed loop control system incorporating the self-tuning controller is shown in Figure 3.
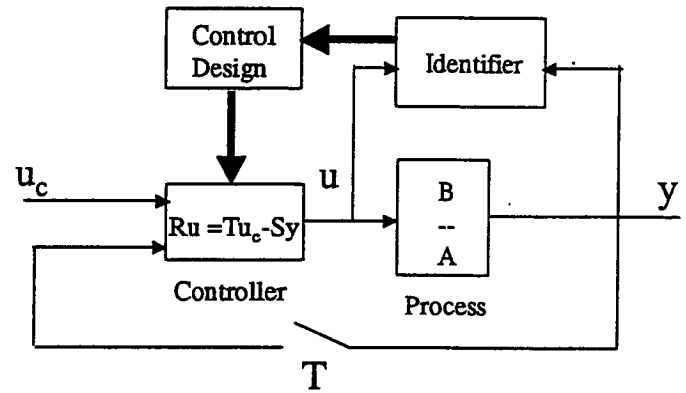

Figure 3 Closed Loop control system with self-tuning controller[7]

When the controlled process is sampled with sampling period T, it can be modeled as;

$$y(k) = \sum_{i=0}^{n} b_i u(k-i) - \sum_{i=1}^{n} a_i y(k-i)$$

The Identifier estimates the a_i and b_i coefficients which are used to calculate the r_i, t_i and s_i coefficients of the controller.
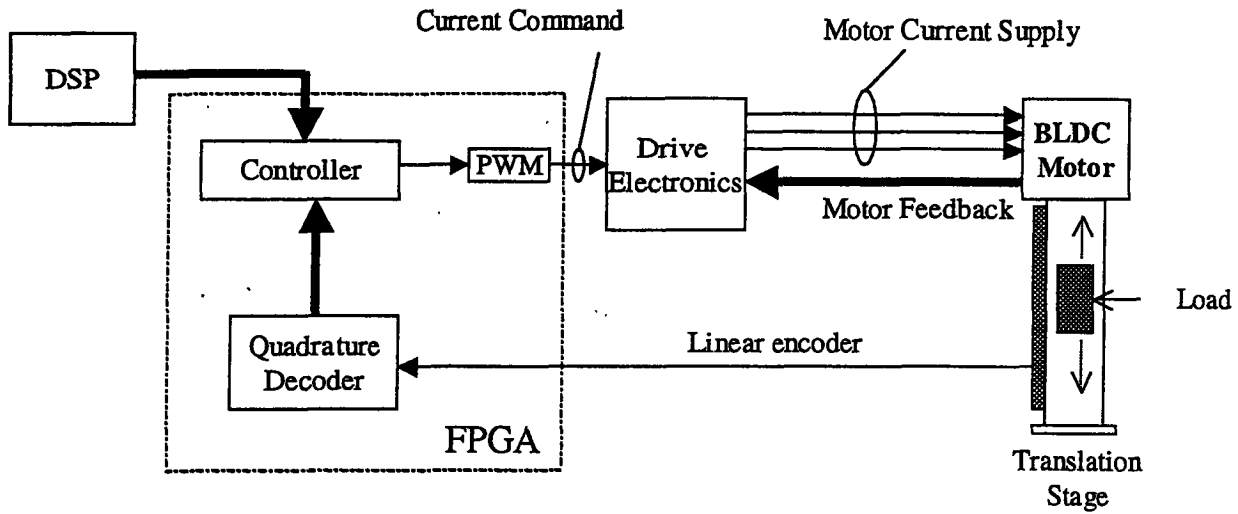
Figure 4 Brushless DC motor control experimental configuration

When n = 2, the controller can be represented as a second order filter with an added set of terms for u_c

$$u(k) = t_0 u_c(k) + t_1 u_c(k-1) - s_0 y(k) - s_1 y(k-1)$$
$$- s_2 y(k-2) - r_1 u(k-1) - r_2 u(k-2)$$

Where the relationship between the controller coefficients r_i, t_i and s_i and the estimated process parameters a_i and b_i is determined by the selection of a desired closed loop response and the pole placement design process. The interested reader is referred to [7] for an explanation of the pole placement design process. The identifier uses the recursive least-squares algorithm to estimate the process coefficients. The model for the process can be represented by

$$y(k) = \Phi^T(k-1)\Theta$$

For a second order system,

$$\Theta^T = \begin{bmatrix} a_1 & a_2 & b_0 & b_1 \end{bmatrix}$$
$$\Phi^T = \begin{bmatrix} -y(k-1) & -y(k-2) & u(k) & u(k-1) \end{bmatrix}$$

The least squares estimator with exponential forgetting is

$$\hat{\Theta}(k) = \hat{\Theta}(k-1) + K(k)\varepsilon(k)$$
$$\varepsilon(k) = y(k) - \Phi^T(k-1)\hat{\Theta}(k-1)$$
$$K(k) = P(t-1)\Phi(k-1)\left(\lambda + \Phi^T(k-1)P(k-1)\Phi(k-1)\right)^{-1}$$
$$P(k) = \left(I - K(k)\Phi^T(k-1)\right)P(k-1)/\lambda$$

This set of equations is used to update the estimates of a_1, a_2, b_0 and b_1 using sampled process outputs and controller inputs.

It is also possible to substitute the PID control equation for the general linear control equation used in the self-tuning controller structure in Figure 3. The PID gains can be tuned on-line using a design approach that makes use of the estimated plant coefficients. Dominant pole design is one such approach [25] and direct synthesis is another design approach (this is an internal model control (IMC) technique) [26].

III. EXPERIMENTAL EMA SYSTEM

The experimental system chosen is a brushless DC (BLDC) motor-driven vertical stage. A diagram of the system is shown in Figure 4. The controller must produce a motor current (torque) command, which is supplied to the motor driver. The motor driver controls the motor current to produce motor torque proportional to the current command. The BLDC motor drives a ballscrew that translates the rotation of the shaft to linear displacement of a carriage. The carriage position is measured using a linear encoder that provides quadrature encoded logic-level signals. A quadrature signal decoder uses these signals to determine the direction of motion and to produce pulses that can be counted to calculate carriage position. The position measurement is provided to the controller to compare with the desired position trajectory. The controller will produce the current command by using the error between the desired position and the measured position. Adding or removing weights can vary the load on the carriage. Since the stage motion is vertical, this provides a variable load torque bias to the motor.
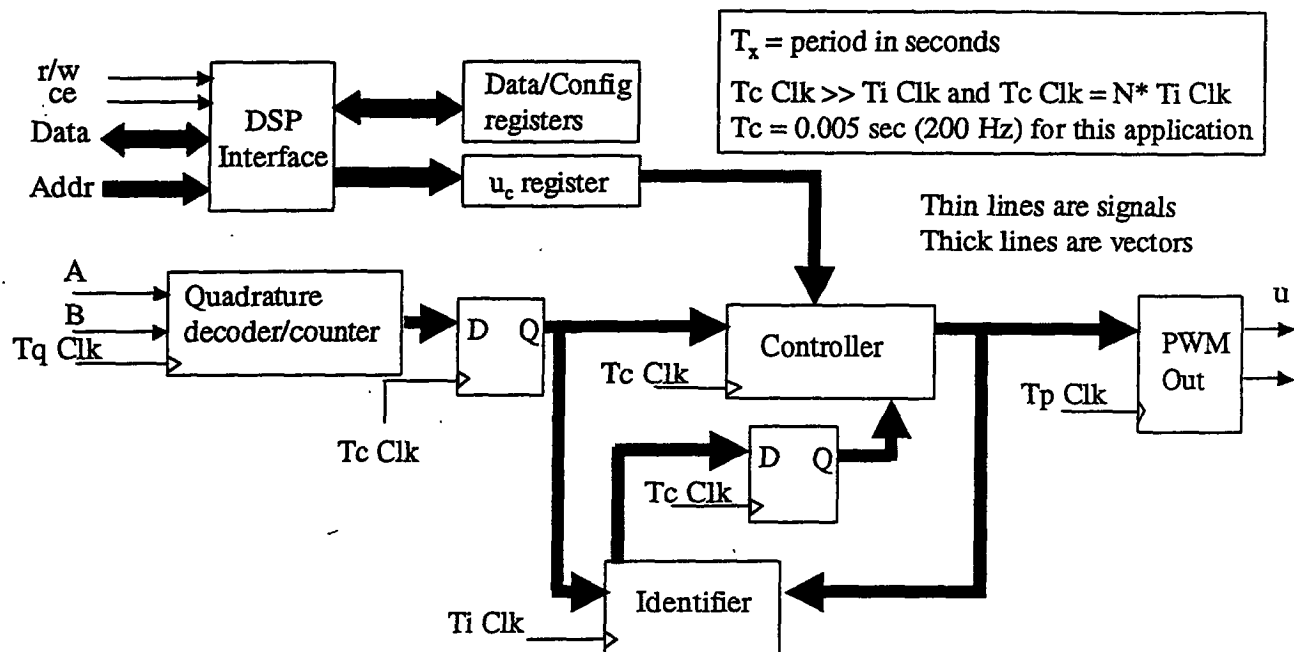
Figure 5 Block diagram of the FPGA implementation of a self-tuning controller

In the figure:

T_x = period in seconds

Tc Clk >> Ti Clk and Tc Clk = N* Ti Clk

Tc = 0.005 sec (200 Hz) for this application

Thin lines are signals
Thick lines are vectors

IV. SELF-TUNING CONTROLLER IMPLEMENTATION

Controller structures utilizing the digital filter representation can easily be implemented in an FPGA as this representation is simply a multiply-accumulate operation. However, for the self-tuning controller using pole placement design, the estimator and the design functions require multiplication and division and are significantly more complicated. The control design adds nine equations with one division and several multiply-accumulate operations each. The identifier adds thirteen equations with multiply accumulate operations and adds eight division operations.

The controller requires peripheral functions to condition sensor measurement and control input to the plant. Most of the required peripheral functions can be implemented as digital circuits on an FPGA. Figure 4 shows the signal conditioning functions are a quadrature decoder/counter for position measurement and a PWM generator for the motor current command. Initially, in the experimental setup, an interface to a DSP will be used to generate desired trajectory commands and for configuration of internal registers. The DSP will also be used to capture internal data pertinent to the analysis of the control system performance. The proposed FPGA implementation of the controller is shown in Figure 5.

Currently the DSP Interface, Quadrature Decoder/Counter and PWM generator functions have been written in VHDL and implemented on a Xilinx Virtex XCV300 FPGA. The DSP interface, quadrature decoder, and three PWM generators consume 17% of the available logic resources and 17% of the available I/O blocks. These functions have been used in the control of the experimental system via a controller implemented in software on the DSP. VHDL coding and testing of both a PID controller and a pole-placement controller is in progress. The design and coding of the identifier and controller design blocks is underway.

V. SUMMARY

Implementation of controllers as digital filters in FPGA's is shown to be feasible, and the implementation of the added components for an adaptive controller is expected to also be successful. The relevant literature contains many examples of successful controller implementations in FPGA's. High order digital filters, with greater complexity than most digital controllers, have also been implemented on FPGA's, implying that controllers of greater complexity can be successfully implemented. The capability of FPGA's to operate at clock rates in the range of 10's to 100's of MHz provides plenty of processing capability. Controllers generally operate at sampling frequencies of much less than 10 kHz, for most practical applications, so throughput should not be a problem. Implementation of the identifier and controller design calculations will be challenging, but is considered feasible. The identifier and design equations have multiple variable terms and will require careful scaling. For deployment, the DSP interface will be replaced by a suitable communication interface for command and control.

REFERENCES

[1] Goslin, G. R., "A Guide to Using Field Programmable Gate Arrays (FPGAs) for Application-Specific Digital Signal Processing Performance", Xilinx Corporation, 1995

[2] Dick, C. and Harris F. J., "Configurable Logic for Digital Communications: Some Signal Processing Perspectives", *IEEE Communications Magazine*, pp.107-111, August 1999

[3] Moeller, T. J. and Martinez, D. R.,"Field Programmable Gate Array Based Radar Front-End Digital Signal Processing", *Proc. 7th Annua IEEE Symp on Field-Programmable Custom Computing Machines*, pp. 178-187, 1999

[4] Kaluri, K.; Wen Fung Leong; Kah-Howe Tan; Johnson, L.; Soderstrand,M.," FPGA hardware implementation of an RNS FIR digital filter", Conference Record of the Thirty-Fifth Asilomar Conf. on Signals, Systems and Computers, 2001., Vol. 2 ,pp. 1340 –1344, 2001

[5] Mishra, A.; Hubbard, A.E., A cochlear filter implemented with a field-programmable gate array, *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, Volume: 49 Issue: 1 , pp. 54 - 60 Jan. 2002

[6] Yamada, M.; Nishihara, A. , High-speed FIR digital filter with CSD coefficients implemented on FPGA , *Proceedings of the Asia and South Pacific Design Automation Conf.*, pp 7-8, 2001

[7] Astrom, K.J., and Wittenmark, B., *Adaptive Control, 2nd ed*, Addison-Wesley Publishing Co. Inc., New York, NY, 1995

[8] Raimondi, G. M., et. al., *Large Electromechanical Actuation Systems for Flight Control Surfaces*, IEE Colloquium on All Electronic Aircraft, 1998

[9] Bolognani, et. al., *Experimental Fault-Tolerant Control of a PMSM Drive*, IEEE Transactions on Industrial Electronics, Vol. 47, No. 5, October 2000.

[10] Croke, S and Herrenschmidt, J., "More Electric Initiative Power-by-wire Actuation Alternatives", *Proceedings of the Aerospace and Electronics Conference*, Vol. 2 , pp. 1338-1346, 1994

[11] Ellis, G., *Control System Design Guide*, Academic Press, 2000

[12] Leonhard, Werner, *Control of Electric Drives, 2nd Edition*, Springer-Verlag, Berlin, Germany, 1997

[13] TMS320F2812 Datasheet, Texas Instruments, 2002

[14] ADSP-21990 Datasheet, Analog Devices, 2002

[15] DSP56F807 Datasheet, Motorola, 2002

[16] Gwaltney, David A., "Test Platform for Advanced Digital Control of Brushless DC Motors (MSFC Director's Discretionary Fund Final Report, Project No. 00-04)", NASA/TM-2002-211917, July 2002

[17] Chen, Ruei-Xi, Chen, Liang-Gee and Chen, Lilin, "System Design Consideration for Digital Wheelchair Controller", *IEEE Transactions on Industrial Electronics*, Vol. 47, No. 4, pp. 898-907, August 2000

[18] Petko, Macej, and Tadeusz, Uhl, "Embedded Controller Design – Mechatronic Approach", *Proceedings of the Second International Workshop on Robot Motion and Control*, pp. 195-200, October 2001

[19] Kim, Se Jin, Lee, Ho Jae, Kim, Sang Koon, Kwon, Young Ahn, "ASIC Design for DTC Based Speed Control of an Induction Motor", *Proceedings on the 2001 IEEE International Symposium on Industrial Electronics, Vol. 2*, pp. 956-961, 2001

[20] Garbergs, B., and Sohlberg, B., "Implementation of a State Space Controller in a FPGA", *Proceedings of the 9th Mediterranean Electrotechnical Conference*, Vol. 1, pp566-569, 1998

[21] Garbergs, B., and Sohlberg, B., "Specialized Hardware for State Space Control of a Dynamic Process", *Proceedings of the IEEE Region 10 International Conference on Computers, Communications, Control and Power Engineering (TENCON)*, Digital Signal Processing Applications , Vol. 2 , 895-899, 1996

[22] Cirstea, Marcian and Dinu, Andrei, "A New Neural Networks Approach to Induction Motor Speed Control", *Proceedings of the 32nd Annual Power Electronic Specialists Conference*, Vol 2, pp 784-787, 2001

[23] Cirstea, Marcian, Khor, Jean and McCormick, Malcolm, "FPGA Fuzzy Logic Controller for Variable Speed Generators", *Proceedings of the IEEE International Conference on Control Applications*, pp 301-304, September 2001

[24] Phillips, Charles L. and Nagle, H. Troy, *Digital Control System Analysis and Design*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984

[25] Astrom, K. and Hagglund, T., *PID Controllers:Theory, Design and Tuning, 2nd ed.*, Instrument Society of America, Research Triangle Park, NC, 1995

[26] Huang, H.-P., Roan, M.-L., Jeng, J.-C., "On line Adaptive Tuning for PID Controllers", IEE Proceedings-Control Theory Applications, Vol.149, No. 1, pp. 60-67, January 2002